

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-182220

(43)Date of publication of application : 21.07.1995

(51)Int.Cl.

G06F 12/00

(21)Application number : 05-322334

(71)Applicant : HITACHI LTD

(22)Date of filing : 21.12.1993

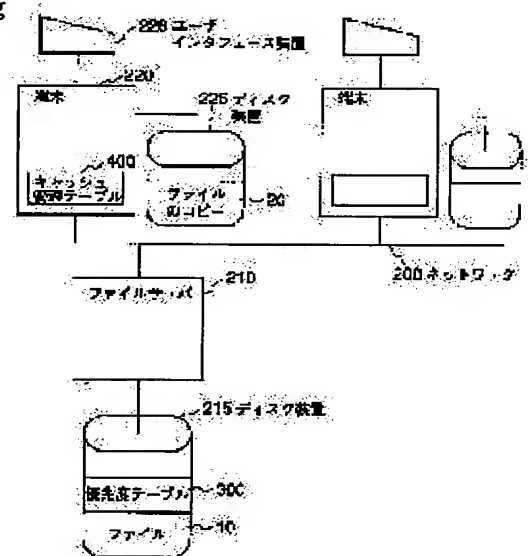
(72)Inventor : NAKANO HIROHIKO
DOMYO SEIICHI
KURODA SAWAKI
SHOJI TADASHI
KOBAYASHI ATSUSHI

(54) DISTRIBUTED FILE SYSTEM AND ITS FILE CACHING METHOD

(57)Abstract:

PURPOSE: To provide a file caching method capable of attaining a high cache hit rate.

CONSTITUTION: A terminal 220 caches files stored by a file server 210 in a disk device 225. A cache management table 400 sorts the files in caching to a 1st group for purging them in the ascending order of priority and a 2nd group for purging them in the older order of final reference time. When a sufficient large space area does not exist in the device 225 in the case of caching a new file, which file out of files belonging to the 1st and 2nd groups is to be purged is selected. A high cache hit rate can be attained by combining a method for purging a file having older final reference time and a method for purging a file having lower Priority.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-182220

(43) 公開日 平成7年(1995)7月21日

(51) Int.Cl.⁶

G 0 6 F 12/00

識別記号

5 4 5 B 8944-5B

Z 8944-5B

片内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数11 O L (全 10 頁)

(21) 出願番号

特願平5-322334

(22) 出願日

平成5年(1993)12月21日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 中野 裕彦

神奈川県川崎市麻生区王禅寺1099 株式会

社日立製作所システム開発研究所内

(72) 発明者 道明 誠一

神奈川県川崎市麻生区王禅寺1099 株式会

社日立製作所システム開発研究所内

(72) 発明者 黒田 澤希

神奈川県川崎市麻生区王禅寺1099 株式会

社日立製作所システム開発研究所内

(74) 代理人 弁理士 富田 和子

最終頁に続く

(54) 【発明の名称】 分散ファイルシステムおよびそのファイルキャッシング方法

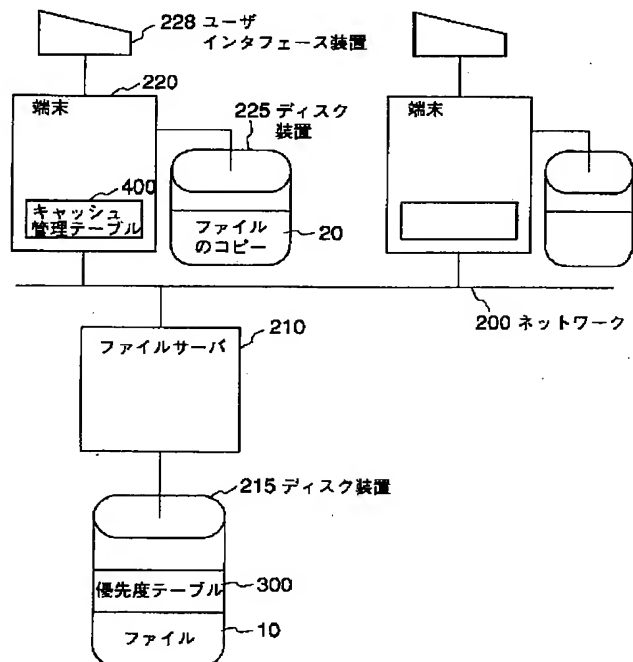
(57) 【要約】 (修正有)

【目的】 高いキャッシュヒット率を達成するファイルキャッシング方法を提供する。

【構成】 端末220は、ファイルサーバ210が保管するファイルをディスク装置225にキャッシングする。キャッシュ管理テーブル400は、キャッシング中のファイルを、優先度が低い順にパージする第1のグループと、最終参照時刻が古い順にパージする第2のグループとに分類する。新たなファイルをキャッシングするにあたり、ディスク装置225に十分な空き領域がない場合は、第1のグループに属するファイルをパージするか第2のグループに属するファイルをパージするかを選択する。

【効果】 最終参照時刻が古いファイルをパージする方法と、優先度が低いファイルをパージする方法を組み合わせることで、高いキャッシュヒット率を達成できる。

第1の実施例 (図2)



【特許請求の範囲】

【請求項 1】 ファイルを保管する第 1 のデータ蓄積装置を有するファイルサーバと、該ファイルサーバ内に保管されたファイルのコピーを保管する第 2 のデータ蓄積装置を有する端末とが、ネットワークによって相互に接続され、前記端末で実行されているアプリケーションがファイルをアクセスする際、該ファイルのコピーが第 2 のデータ蓄積装置に存在する場合は、当該ファイルのコピーにアクセスする分散ファイルシステムにおけるファイルキャッシング方法において、前記端末において、第 2 のデータ蓄積装置に存在するファイルのコピーを第 1 および第 2 のグループに分類し、第 1 のグループに属するコピー間では、各コピーに設定された優先度に従って第 1 のページ順序を付けるとともに、第 2 のグループに属するコピー間では、最終参照時刻が古い順に第 2 のページ順序を付け、新たなファイルのコピーを第 2 のデータ蓄積装置に保管するにあたり、そのコピーを保管するのに必要な空き領域を確保するために、既に第 2 のデータ蓄積装置に保管されている他のコピーをページする必要がある時に、前記第 1 のグループに属するコピーを前記第 1 のページ順序に従ってページするか、第 2 のグループに属するコピーを前記第 2 のページ順序に従ってページするかを選択することを特徴とするファイルキャッシング方法。

【請求項 2】 前記第 1 のグループが空でない場合は、そのグループに属するコピーを前記第 1 のページ順序に従ってページし、前記第 1 のグループが空の場合は、前記第 2 のグループに属するコピーを前記第 2 のページ順序に従ってページすることを特徴とする請求項 1 記載のファイルキャッシング方法。

【請求項 3】 前記ファイルサーバが当該ファイルサーバに保管された各ファイルの優先度を管理しており、任意のファイルのコピーを前記第 2 のデータ蓄積装置に保管する際、当該優先度を、前記第 1 のページ順序の決定に関与するコピーの優先度として用いることを特徴とする請求項 1 記載のファイルキャッシング方法。

【請求項 4】 前記端末が前記ファイルサーバに保管された各ファイルの優先度を管理しており、任意のファイルのコピーを前記第 2 のデータ蓄積装置に保管する際、前記優先度テーブルを参照して当該コピーの優先度を設定することを特徴とする請求項 1 記載のファイルキャッシング方法。

【請求項 5】 前記ファイルの優先度を、そのファイルへのアクセス頻度に基づいて決定することを特徴とする請求項 3 または 4 記載のファイルキャッシング方法。

【請求項 6】 前記第 1 および第 2 のグループの分類の際、前記第 2 のデータ蓄積装置に保管された各ファイルのコピーについて、その最終参照時刻からの経過時間がタイムアウト時間 T より大きいコピーを前記第 1 のグループに、前記タイムアウト時間 T より小さいコピーを前

記第 2 のグループに分類することを特徴とする請求項 1 または 2 記載のファイルキャッシング方法。

【請求項 7】 一連のファイルアクセスをおこなうジョブが終了した時に、前記第 2 のデータ蓄積装置に存在するファイルのコピーをすべて前記第 1 のグループに所属させることを特徴とする請求項 1 記載のファイルキャッシング方法。

【請求項 8】 実行中のジョブがアクセスしたコピーは前記第 2 のグループに所属させ、アクセスしていないコピーは、前記第 1 のグループに所属させることを特徴とする請求項 1 記載のファイルキャッシング方法。

【請求項 9】 ファイルを固定長ブロックに分割して、ブロック単位にファイルのコピーを第 2 のデータ蓄積装置に保管し、ブロック単位にファイルのコピーを前記第 2 のデータ蓄積装置からページすることを特徴とする請求項 1 記載のファイルキャッシング方法。

【請求項 10】 同一ファイルに属するブロック間では、ファイルの先頭に近いブロックほど、高い優先度を設定することを特徴とする請求項 9 記載のファイルキャッシング方法。

【請求項 11】 ファイルを保管する第 1 のデータ蓄積装置を有するファイルサーバと、該ファイルサーバ内に保管されたファイルのコピーを保管する第 2 のデータ蓄積装置を有する端末とが、ネットワークによって相互に接続され、前記端末で実行されているアプリケーションがファイルをアクセスする際、該ファイルのコピーが第 2 のデータ蓄積装置に存在する場合は、当該ファイルのコピーにアクセスする分散ファイルシステムにおいて、前記端末は、第 2 のデータ蓄積装置に存在するファイルのコピーを第 1 および第 2 のグループに分類し、第 1 のグループに属するコピー間では、各コピーに設定された優先度に従ってページ順序を付け、第 2 のグループに属するコピー間では、最終参照時刻が古い順にページ順序を付けたキャッシュ管理手段を有し、前記ファイルサーバから新たなファイルのコピーを前記第 2 のデータ蓄積装置に保管するにあたり、そのコピーを保管するのに必要な空き領域を確保するために、既に第 2 のデータ蓄積装置に保管されている他のファイルのコピーをページする必要がある時に、第 1 のグループに属するコピーをページ順序に従ってページするか、第 2 のグループに属するコピーをページ順序に従ってページするかを選択する選択手段を有することを特徴とする分散ファイルシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ファイルを保管するファイルサーバと、ファイルを利用する端末とがネットワークで相互に接続された分散ファイルシステムにおけるファイルキャッシング方法に関する。

【0002】

【従来の技術】分散ファイルシステムは、ファイルを保管するホスト計算機（以下、ファイルサーバと呼ぶ）と、このファイルを利用する端末とがネットワークで結合した構成を持つ。端末は、ファイルサーバが保管するファイルをネットワークを通してアクセスする。

【0003】ファイルをファイルサーバに集め、複数の端末で共有することにより、ファイルのバックアップや複数ユーザ間のファイル共用が容易となる。しかし、ネットワークを経由するため、ファイルアクセスに時間がかかるという問題点がある。

【0004】多くの分散ファイルシステムは、一度アクセスしたファイルのコピーを端末の主記憶やローカルディスクに保持（キャッシング）し、2回目以降の同一のファイルへのアクセスを端末内で行なうことにより、ファイルアクセスの高速化を図っている。例えば、AFS（Andrew File System）は、ファイル全体をローカルディスクにキャッシングする。NFS（Network File System）は、ファイルを固定長ブロック単位に主記憶上にキャッシングする。AFS、NFSについては、エー・シー・エム、コンピューティングサーベイズ、第22巻、第4号（1990年）第321～374頁（ACM, Computing Surveys, Vol. 22, No. 4（1990）pp321-374）に記述されている。

【0005】

【発明が解決しようとする課題】通常、キャッシュ領域の容量には制限があり、アクセスしたすべてのファイルをキャッシングできない。ファイルをキャッシングするにあたり、そのファイルに対して十分な空間を確保できない場合には、先にキャッシングしているファイルをキャッシュ領域からパージする必要がある。キャッシュのヒット率を高め、ファイルアクセス性能を向上するためには、最も遠い将来までアクセスされないファイルをパージすることが望ましい。

【0006】多くの分散ファイルシステムでは、最も長期間アクセスされていないファイルをパージするLRU（Least Recently Used）法を採用している。LRU法は、もともと仮想記憶システムにおけるページ置き換え方法として考案されたものである。LRU法の根拠は、プログラムのページ参照を観測すると、一般にある時間範囲では参照するページの集合は局所化しているという性質（局所参照性）に基づいている。端末で実行されるジョブのファイルアクセスパターンにも局所参照性が存在するという仮定から、LRU法はファイルのキャッシングにも有効であると考えられている。

【0007】このように、LRU法は、ジョブのファイルアクセスパターンに関する経験則に基づいている。

【0008】しかしながら、実際のファイルのアクセスの仕方は、ファイルをアクセスする部署、ユーザあるい

は状況によって様々であり、必ずしも上記の方法によって最適なファイルキャッシングが行なえるとは限らない、という問題があった。

【0009】従って、本発明の目的は、高いキャッシュヒット率を達成することができる分散ファイルシステムおよびそのキャッシング方法を提供することにある。

【0010】本発明の他の目的は、ファイルをアクセスする部署、ユーザ、状況等によらず、高いキャッシュヒット率を達成することができる分散ファイルシステムおよびそのファイルキャッシング方法を提供することにある。

【0011】

【課題を解決するための手段】上記目的を達成するために、本発明のファイルキャッシング方法では、キャッシング中のファイルを、予想アクセス頻度が低い順にパージする第1のグループと、最終参照時刻が古い順にパージする第2のグループとに分類する。本発明の望ましい形態では、最終参照時刻からの経過時間がタイムアウト時間Tより大きなファイルを第1のグループに、タイムアウト時間Tより小さなファイルを第2のグループに所属させる。

【0012】新たなファイルをキャッシングするにあたり、そのファイルのコピーを保管するのに十分なキャッシュ領域を確保できない場合は、第1のグループに属するファイルをパージする。第1のグループが空の場合は、第2のグループに属するファイルをパージする。

【0013】また、この方法を実施するシステムとして、本発明による分散ファイルシステムは、ファイルを保管する第1のデータ蓄積装置を有するファイルサーバと、該ファイルサーバ内に保管されたファイルのコピーを保管する第2のデータ蓄積装置を有する端末とが、ネットワークによって相互に接続され、前記端末で実行されているアプリケーションがファイルをアクセスする際、該ファイルのコピーが第2のデータ蓄積装置に存在する場合は、当該ファイルのコピーにアクセスする分散ファイルシステムにおいて、前記端末は、第2のデータ蓄積装置に存在するファイルのコピーを第1および第2のグループに分類し、第1のグループに属するコピー間では、各コピーに設定された優先度に従ってパージ順序を付け、第2のグループに属するコピー間では、最終参照時刻が古い順にパージ順序を付けたキャッシュ管理手段を有し、前記ファイルサーバから新たなファイルのコピーを前記第2のデータ蓄積装置に保管するにあたり、そのコピーを保管するのに必要な空き領域を確保するために、既に第2のデータ蓄積装置に保管されている他のファイルのコピーをパージする必要が生じた時に、第1のグループに属するコピーをパージ順序に従ってパージするか、第2のグループに属するコピーをパージ順序に従ってパージするかを選択する選択手段を有するものである。

【0014】

【作用】実行中のジョブの局所参照性がある場合には、前述のようにLRU法によるファイルコピーのページ順決定が有効である。

【0015】LRU法の他に、ジョブに局所参照性がない場合には、過去のアクセス頻度からページするファイルを決定する方法も考えられる。この根拠は、頻繁にアクセスされたファイルは近い将来にも高い確率でアクセスされるという仮定に基づいている。

【0016】しかし、実行中のジョブに局所参照性がある場合でも、アクセス頻度に基づいてページする順序を決定する方法が有効である場合がある。例えば、実行中のジョブがアクセスしないファイル間では、最終参照時刻によらずアクセス頻度に基づいてページする順序を決定する方法が適している。また、実行中のジョブがアクセスしたファイルであっても、長期間アクセスされていないファイルは局所参照しているファイルの集合から外れたとみなせる。これらのファイル間では、最終参照時刻よりもアクセス頻度に基づいてページする順序を決定する方が好ましい。ジョブが終了し、これまでのファイルアクセスパターンが参考にならない場合にも、アクセス頻度に基づく方法は有効である。

【0017】本発明者らは、以上の考察から、LRU法とファイルのアクセス頻度に基づく方法とを上述のように組み合わせることで、更にキャッシュのヒット率を向上できることに想到した。

【0018】本発明の具体的構成においては、長期間アクセスされていないファイルは第1のグループに所属する。それ以外のファイルは第2のグループに所属する。新たなファイルをキャッシングするにあたり、そのファイルのコピーを保管するのに十分なキャッシュ領域を確保できない場合は、いずれかのファイルコピーのページが必要となる。

【0019】長期間アクセスされていないファイルが存在しない場合、すなわち第1グループに属するファイルが存在しない場合、第2グループのファイルの中から最も参照時刻の古いファイルをページ対象とする。第1のグループに属するファイルが存在すれば、それらのファイルうちで、優先度（例えばアクセス頻度）が最も低いものをページ対象とする。すなわち、一定期間以上アクセスされていないファイルについては、それらの最終参照時刻の差は問題とせず、アクセス頻度のみを考慮とする。

【0020】例えば、端末がアイドル状態にあって、長期間ファイルアクセスが発生しない状況を考える。このような状況では、最終参照時刻よりもアクセス頻度の方が、近い将来にアクセスされる可能性を予測する指標として適している。すなわち、最も長期間アクセスされていないファイルをページするLRU法より、長期間アクセスされていないファイルのうちでアクセス頻度が最も

低いファイルをページする方法の方が、高いキャッシュヒット率を達成できる。

【0021】

【実施例】以下、本発明の好適実施例について、図面により詳細に説明する。

【0022】（実施例1）まず、第1の実施例について説明する。

【0023】図2は、本発明を適用した分散ファイルシステムの構成を示している。ネットワーク200には、ファイルサーバ210と複数の端末220とが接続されている。本実施例では、ネットワーク200としては、ローカルエリアネットワークを採用する。ファイルサーバ210には高性能ワークステーションを、端末220には安価なワークステーション或はパーソナルコンピュータを採用する。

【0024】ファイルサーバ210には、大容量（数百メガバイト以上）のディスク装置215が接続している。各端末220には、小容量（数十メガバイト程度）のディスク装置225と、キーボードやディスプレイを備えたユーザインタフェース装置228が接続している。また、各端末220は、図示しないが、時計機能を有し、ファイルの最終参照時刻からの経過時間を計測することができる。

【0025】ファイルサーバ210は、付属のディスク装置215に、多数のファイル10と、優先度テーブル300とを保管している。優先度テーブル300については後述する。端末220は、付属のディスク装置225に、ファイル10のコピー20を保管する。

【0026】図3は、優先度テーブル300の構造を示している。優先度テーブル300は、ファイル名欄310と優先度欄320からなり、ファイルサーバ210が保管する各ファイル10に対して、そのファイル名と優先度とを対応づける。優先度欄320の優先度の値は、ファイル10に対する予想されるアクセス頻度に基づいて、システム管理者（またはユーザ）が決定する。あるいは、実際の過去のアクセス頻度情報に基づいてシステム管理者が決定するようにしてもよい。図の例では、アクセス頻度が高いファイルほど、高い優先度が設定され、高い優先度をもつファイルほどページの対象とらしくなる。

【0027】各ファイルに対してデフォルトの優先度を設定し、その他の優先度を持つファイルだけを優先度テーブル300に登録する方法も考えられる。また、ファイルサーバ210において、過去のアクセスを計数し、システム管理者の代わりに優先度を更新設定するようにすることも可能である。

【0028】図4は、端末220内のキャッシュ管理テーブル400の構造を示している。優先度構造体410は、ファイル名欄411、優先度欄412、最終参照時刻欄413、アドレス欄414、ポインタ欄415から

なる構造体であり、ディスク装置225に存在するファイルのコピー20毎に1つ作成される。ポインタ欄415は、後述するリスト構造を構成するために使用する。

【0029】各ファイルについて定められた個々の優先度構造体410は、第1のグループ470と第2のグループ480とのいずれかに分類され、それぞれ第1リスト420、第2リスト430につながれる。第1リスト420は、優先度欄412の値が小さい（すなわちアクセス頻度が低い）順に優先度構造体410を整列する。第2リスト430は、最終参照時刻欄413の値（時刻）が古い順に優先度構造体410を整列する。

【0030】次に、各端末220における、キャッシュ管理テーブル400に対する操作を説明する。

【0031】（1）登録操作

新たにキャッシングされたファイルに対して、その優先度構造体410を作成する。この際、第2リスト430の末尾につなぐ。ファイル名欄411には、登録するファイル10の名前を設定する。優先度欄412には、優先度テーブル300に登録されているファイル10の優先度320を設定する。最終参照時刻欄413には、現時刻を設定する。アドレス欄414には、コピーが存在するディスク装置225内のアドレスを設定する。

【0032】（2）削除操作

パージされたファイルに対して、その優先度構造体410をリストから外し、削除する。

【0033】（3）最終参照時刻の更新操作

すでにキャッシングされているファイルに対して再度アクセスがあった場合、そのファイルの優先度構造体410について最終参照時刻欄413に現時刻を設定し、優先度構造体410を第2リスト430の末尾に移動する。

【0034】（4）タイムアウト判定操作

端末220は、定期的に（例えば、1分おきに）、第2リスト430を探索し、最終参照時刻413からの経過時間が、予め定めたタイムアウト時間Tより大きくなった優先度構造体410を第1リスト420に移動する。その際、そのファイルの優先度に応じたリスト中の位置に配置する。

【0035】これによって、最終参照時刻413からの経過時間がタイムアウト時間T（例えば、1時間）より大きなファイルの優先度構造体410が第1のグループに含まれ、タイムアウト時間Tより小さなファイルの優先度構造体410が第2のグループに含まれるよう、分類がなされる。

【0036】端末220は、ユーザによる或いは実行中のプログラムによるファイルアクセス要求を処理する。図5のフローチャートを用いて、端末220のファイルアクセス処理を説明する。まず、アクセスするファイル10のコピーがディスク装置225に存在するかチェックする（ステップ502）。存在しない場合は、ファイ

ル10をディスク装置225にキャッシングする（ステップ504）。次に、ディスク装置225に存在するファイル10のコピー20にアクセスし（ステップ506）、コピー20の最終参照時刻を更新する（ステップ508）。

【0037】次に、図1のフローチャートを用いて、図2のステップ504のファイルキャッシング処理の詳細例を説明する。まず、キャッシングするファイルのサイズをファイルサーバ210に問い合わせる（ステップ102）。次に、この問合せにより得られたサイズのファイルに対して、ディスク装置225に、十分な空き領域があるかチェックする（ステップ104）。十分な空き領域がある場合は、ファイル10のコピーをディスク装置225に保管し（ステップ114）、その優先度構造体410を前述のように生成してキャッシュ管理テーブル400に登録する（ステップ116）。

【0038】ディスク装置225に、十分な空き領域がない場合は、既にキャッシングしているファイルのコピーをパージする。どのファイルのコピーをパージするかは、キャッシュ管理テーブル400を参照して決定する。まず、キャッシュ管理テーブル400の第1リスト420が空であるかチェックし（ステップ106）、空でない場合は、第1リスト420の先頭（優先度が最低）に位置するファイルのコピーをパージする（ステップ108）。第1リスト420が空の場合は、第2リスト430の先頭（最終参照時刻が最古）に位置するファイルのコピーをパージする（ステップ110）。パージしたファイルのコピーをキャッシュ管理テーブル400から削除（ステップ112）した後、ステップ104に戻って、ディスク装置225に十分な空き領域があるか再度チェックする。

【0039】このように、本実施例では、長期間アクセスされていないファイルのコピー20を第1のグループに所属させる。新たなファイル10をキャッシングするにあたり、ディスク装置225に十分な空き領域がない場合は、原則として、第1のグループに属する、すなわち長期間アクセスされていないコピー20のうちで、優先度（すなわちアクセス頻度）が最も低いものをパージする。この際、上記タイムアウト時間T以上アクセスされていないファイル群（第1グループ）については、もはや最終参照時刻の差は考慮されず、優先度の方が考慮されるようになることに留意されたい。

【0040】例えば、端末220がアイドル状態にあって、長期間ファイルアクセスが発生しない状況を考える。このような状況では、すべてのキャッシングされているファイルについてその最終参照時刻からタイムアウト時間Tが経過していると考えられる。このような場合、最終参照時刻よりも過去のアクセス頻度の方が、近い将来にアクセスされる可能性を予測する指標として適している。従って、最も長期間アクセスされていないフ

ファイルのコピー20をパージするLRU法に比べて、長期間アクセスされていないファイルのコピー20のうちで、アクセス頻度が最も低いファイルをパージする方法の方が、高いキャッシュヒット率を達成できる。

【0041】(実施例2)次に第2の実施例について説明する。

【0042】本実施例では、ジョブの終了時に、第2リスト430に所属する優先度構造体410をすべて第1リスト420に移動させるようにする。これにより、実行中のジョブがアクセスしないファイルのコピー20を第1のグループに所属させる。その他の処理は、第1の実施例と同じである。

【0043】最終参照時刻が古い順にパージするLRU法の根拠は、端末220で実行されるジョブのファイルアクセスパターンには局所参照性が存在するという経験則に基づいている。しかし、実行中のジョブがアクセスしないファイル間では、近い将来アクセスされる可能性を予測する指標として、最終参照時刻は参考にならない。むしろ、アクセス頻度に基づいてパージする順序を決定する方が有効であると考えられる。

【0044】従って、ジョブの終了後は、最も長期間アクセスされていないファイルのコピー20をパージするLRU法に比べて、実行中のジョブがアクセスしていないファイルのコピー20のうちで、アクセス頻度が最も低いファイルをパージする方法の方が、高いキャッシュヒット率を達成できる。

【0045】(実施例3)次に、第3の実施例について説明する。

【0046】本実施例では、ファイルを固定長ブロックに分割し、ブロック単位にキャッシングする。更に、分割されたファイルの先頭ブロックに高い優先度を設定することで順次アクセスを高速化する。

【0047】図6は、本実施例における優先度テーブル300の構造を示している。本実施例では、優先度テーブル300に、「順次アクセスフラグ欄」330を追加する。順次アクセスされる可能性が高いファイルに対しては、順次アクセスフラグ欄330をオンに設定する。

【0048】図7は、本実施例におけるキャッシュ管理テーブル400の構造を示している。本実施例では、優先度構造体410に、「ブロック番号欄」416と「順次アクセスフラグ欄」417とを追加している。そして、キャッシング中のファイルのブロック毎に優先度構造体410を割り当てる。

【0049】優先度欄412と順次アクセスフラグ欄417とには、前記優先度テーブル300を参照して、ブロックが属するファイルの優先度310と順次アクセスフラグ320の値(オンまたはオフ)を設定する。ただし、順次アクセスフラグがオンであるファイルの先頭ブロックの優先度は、そのファイルの優先度を+1だけ増加させた値とする。例えば、図7のf001の先頭ブロッ

ック(ブロック番号1)702の優先度“4”は、同じファイルの後続のブロック(ブロック番号2)701の優先度“3”に1を加えたものとなっている。

【0050】図8のフローチャートを用いて、本実施例におけるファイルアクセス処理を説明する。まず、アクセスするファイルブロックのコピーがディスク装置225に存在するかチェックする(ステップ802)。存在しない場合は、ブロックをディスク装置225にキャッシングする(ステップ804)。ディスク装置225に存在するコピーにアクセスし(ステップ806)、コピーの最終参照時刻を更新する(ステップ808)。

【0051】次に、そのファイルの順次アクセスフラグ417がオンか否かチェックし(ステップ810)、オンの場合は、非同期キャッシングプロセスに、次のブロックのキャッシングを依頼する(ステップ812)。そして、次のブロックのキャッシング完了を待つことなく処理を終了する。

【0052】ブロックのキャッシング処理(ステップ804)は、キャッシングの単位が固定長であることを除いて、第1の実施例のファイルキャッシング処理(図1)と同じである。

【0053】本実施例によれば、ファイルの先頭ブロックに高い優先度を設定し、ディスク装置225に残す確率を高めることにより、先頭ブロックのアクセスを高速化できる。2番目以降のブロックは、前のブロックにアクセスしている間にキャッシングが開始されるので、高速にアクセスできる。以上の効果により、ファイルを高速に順次アクセスできる。

【0054】(実施例4)第1の実施例では、ファイルサーバ210が優先度テーブル300を保持し、分散ファイルシステムの管理者が、各ファイルのアクセス頻度情報をもとに各ファイルに対して単一のファイル優先度を設定するものであった。すべての端末220は、この単一の優先度に従って、第1グループのファイル間のパージ順序を決定する。

【0055】しかし、同じファイルでも、部署やユーザによってアクセス頻度が異なる。例えば、プログラム開発者はソースコードを頻繁にアクセスするが、一般ユーザはコンパイルした結果得られる実行プログラムをアクセスし、ソースコードには滅多にアクセスしない。したがって、端末220が設置された部署や、端末220を利用するユーザが、独自にファイルの優先度を設定できることが望ましい。

【0056】図9は、本実施例における分散ファイルシステムの構成を示している。本実施例では、単一のサーバ210に優先度テーブル300を持つのではなく、各々の端末220に優先度テーブル300を持ち、端末毎にファイルの優先度を設定できるようにした。

【0057】端末を利用するユーザは、自分のファイルアクセス傾向に基づいて、ファイルの優先度を決定する

ことができる。例えば、プログラム開発者は、ソースコードに高い優先度を設定し、一般ユーザは、実行プログラムに高い優先度を設定する。これによって、部署やユーザに適したファイルコピーのパージ順を定めることができ、キャッシュヒット率を向上させるが可能になる。

【0058】

【発明の効果】本発明によれば、分散ファイルシステムにおいて、キャッシング中のファイルを、アクセス頻度が低い順にパージする第1のグループと、最終参照時刻が古い順にパージする第2のグループとに分類することにより、従来以上の高いキャッシュヒット率を達成することができる。

【0059】

【図面の簡単な説明】

【図1】本発明の第1の実施例におけるファイルキャッシング処理を示すフローチャート。

【図2】第1の実施例の分散ファイルシステムの構成図。

【図3】第1の実施例における優先度テーブル300の

構成図。

【図4】第1の実施例におけるキャッシュ管理テーブル400の構成図。

【図5】第1の実施例におけるファイルアクセス処理を示すフローチャート。

【図6】第3の実施例における優先度テーブル300の構成図。

【図7】第3の実施例におけるキャッシュ管理テーブル400の構成図。

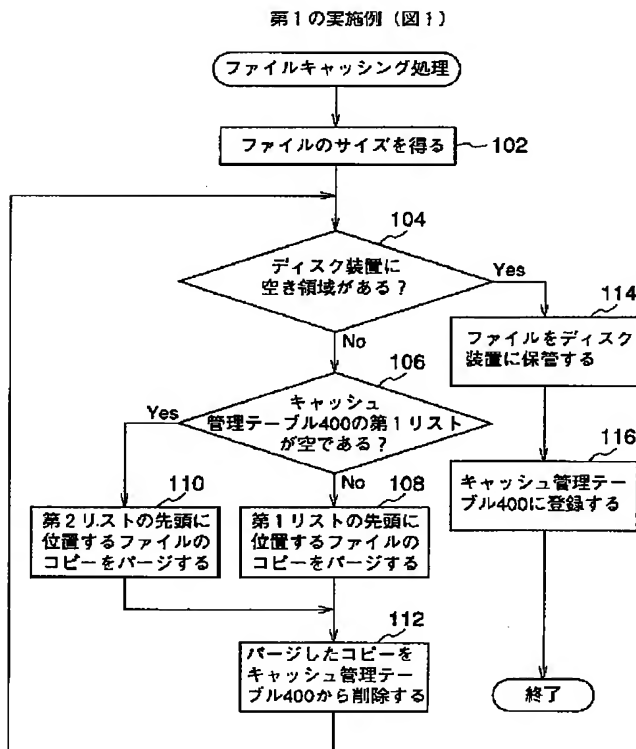
【図8】第3の実施例におけるファイルアクセス処理を示すフローチャート。

【図9】第4の実施例の分散ファイルシステムの構成図。

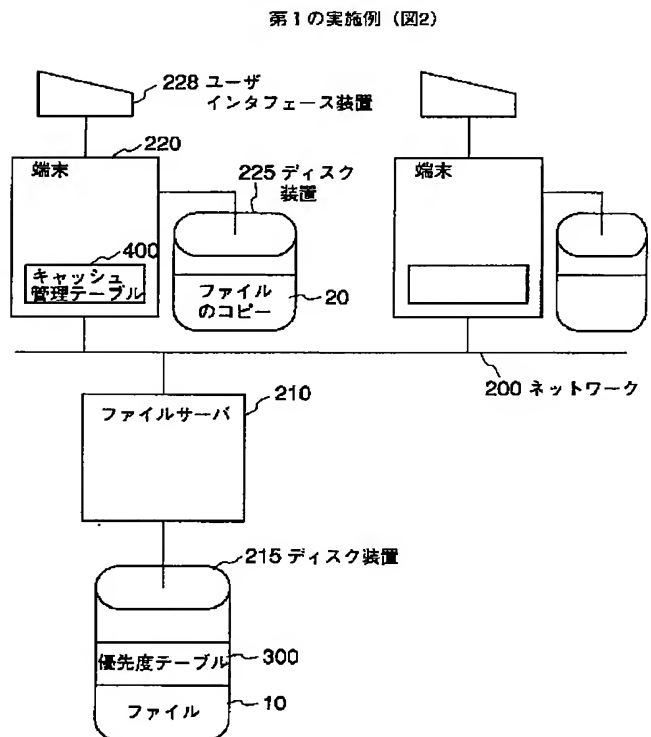
【符号の説明】

200…ネットワーク、210…ファイルサーバ、220…端末、225…ディスク装置、300…優先度テーブル、400…キャッシュ管理テーブル、10…ファイル、20…ファイルのコピー

【図1】



【図2】



【図 3】

図 3

300 優先度テーブル

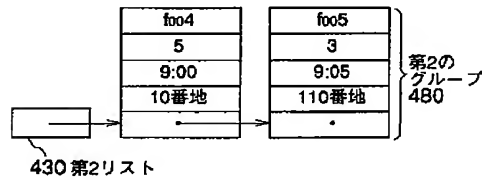
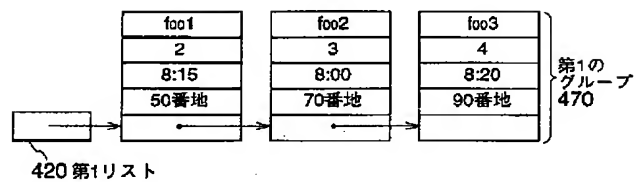
ファイル名欄	優先度欄
foo1	2
foo2	3
foo3	4
foo4	5
foo5	3
⋮	⋮

【図 4】

キャッシュ管理テーブル 400 (図 4)

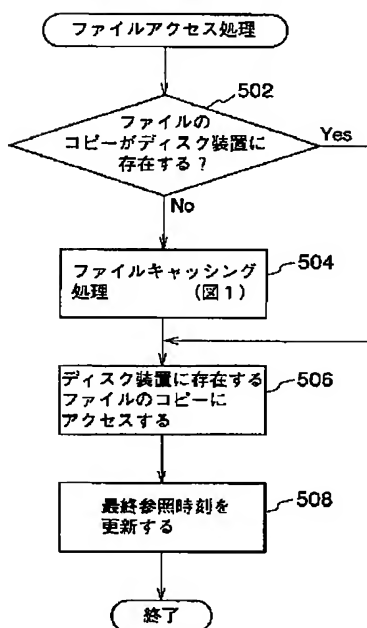
410 優先度構造体

ファイル名欄	411
優先度欄	412
最終参照時刻欄	413
アドレス欄	414
ポインタ欄	415



【図 5】

第1の実施例 (図5)



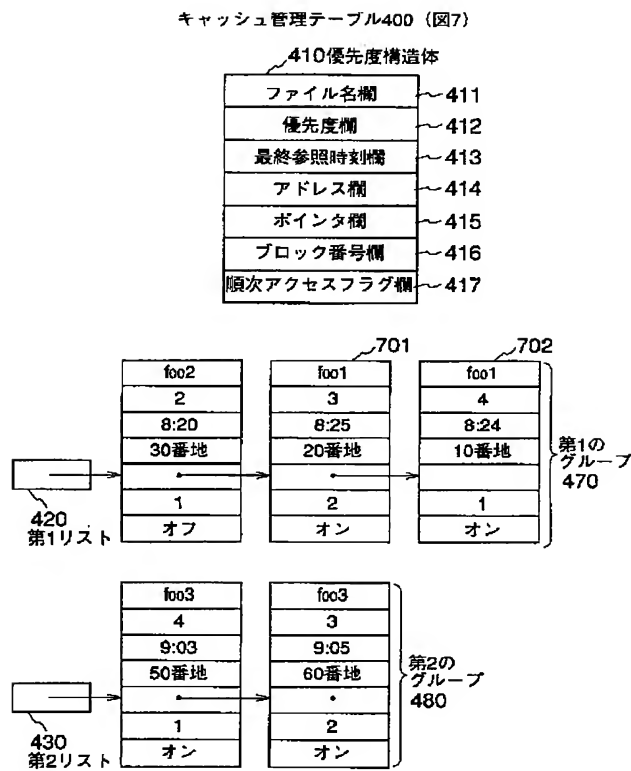
【図 6】

図 6

300 優先度テーブル

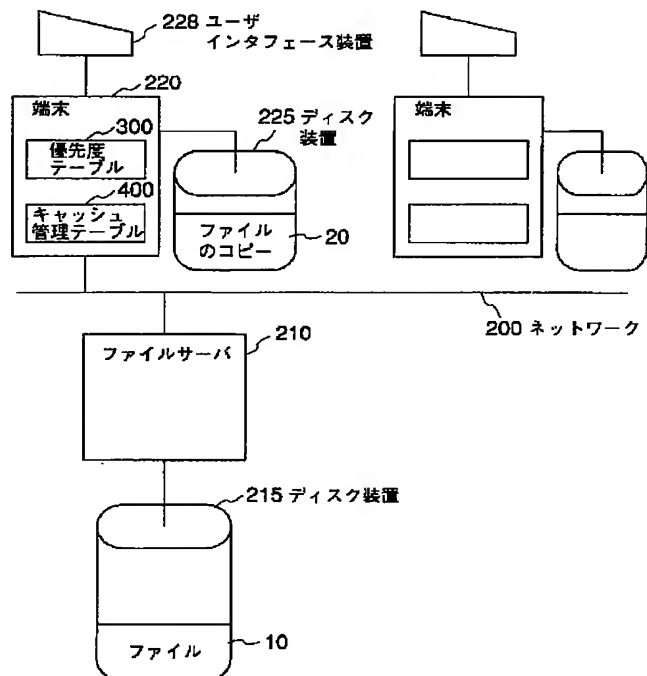
ファイル名欄	優先度欄	順次アクセスフラグ欄
foo1	3	オン
foo2	2	オフ
foo3	3	オン
⋮	⋮	⋮

【図7】



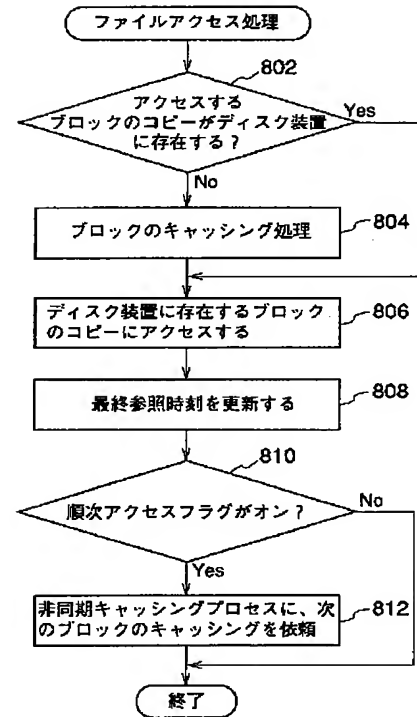
【図9】

第4の実施例 (図9)



【図8】

第3の実施例 (図8)



フロントページの続き

(72)発明者 庄司 直史

神奈川県川崎市麻生区王禅寺1099 株式会
社日立製作所システム開発研究所内

(72)発明者 小林 敦

神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア開発本部内